Office de la propriété intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An Agency of Industry Canada

Bureau canadien des brevets
Certification

Canadian Patent Office
Certification

La présente atteste que les documents ci-joints, dont la liste figure ci-dessous, sont des copies authentiques des documents déposés au Bureau des brevets.

This is to certify that the documents attached hereto and identified below are true copies of the documents on file in the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No: **2,225,227,** on December 18, 1997, by **MICHAEL COVELEY AND VOJIN ZIVOJINOVIC,** for "Intelligent Communication and Applications Server".

Agent certificateur/Certifying Officer

December 2, 2003
Date

INTELLIGENT COMMUNICATION AND APPLICATIONS SERVER

Field of the Invention

The present invention relates to communications systems and in particular to an intelligent communication and applications server to enhance communication connectivity in wireless and/or land-line networks.

Background of the Invention

In today's communications environment, users on different networks must communicate with each other. This makes internetworking among various telecommunications networks an important and challenging technological task. The majority of current applications of internetworking provide access only for selected applications. Difficulties in communication connectivity between applications on wireless networks and host computers on land-line networks has been of concern. Also, the growth of existing and the introduction of new services on communication networks has significantly increased traffic flow. In heterogeneous wired and wireless networks, communication servers must be able to cope with capacity and network failures while being cost efficient. Accordingly, a communication server to provide flexible communication and connectivity between participants is desired.

It is therefore an object of the present invention to provide a novel intelligent communication and applications server to enhance communications connectivity in wireless and/or land-line networks.

Summary of the Invention

Broadly stated, the present invention provides an intelligent communication and applications server to provide communication connectivity between applications such as mobile point-of-sale (POS) / electronic fund transfer (EFT) terminals on a wireless packet data network (WPDN) and host computers on a land-line network. The intelligent communication and applications server wraps API data from the host computers with a logical message which acts as a transporter for

the API data to the destination. In this manner, communication connectivity between the applications on the WPDN and the host computers can be maintained.

The present invention provides advantages in that communication connectivity between the applications on the WPDN and the host computers on the land-line network is maintained even though the API data of the host computers and mobile terminals may not provide for direct communication connectivity. In addition, the present invention provides advantages in that the intelligent communication and applications server includes a knowledge base to deal with "special" communication conditions as they arise.

Brief Description of the Drawings

A preferred embodiment of the present will now be described more fully with reference to the accompanying drawings in which:

Figure 1 illustrates a communications system including an intelligent communication and applications server acting between mobile terminals on a wireless packet data network and host computers on a land-line network in accordance with the present invention;

Figure 2 is a schematic diagram in block form of the intelligent communication and applications server of Figure 1;

Figure 3 illustrates a message dispatcher and a communications manager within one of the mobile terminals of Figure 1;

Figure 4 shows outgoing message flow from the mobile terminal of Figure 3;

Figure 5 shows incoming message flow within the mobile terminal of Figure 3;

Figure 6 illustrates connectivity between the mobile terminals and host computer of the communications system of Figure 1;

Figure 7 illustrates the network infrastructure of the communications system of Figure 1;

Figure 8 is a schematic diagram in block form of the intelligent communication and applications server of Figure 2;

Figures 9 and 10 show communication traffic between the host computers and the mobile terminals; and

Figure 11 shows an OSI model protocol stack and its conversions in the communications chain across the communications system of Figure 1 based on a DataTAC wireless packet data network.

Description of the Invention

In the well known OSI Reference Model, every device that connects two computer systems or networks, that are not connected to each other, is referred to as a relay. In accordance with this terminology, a bridge functions as a data link layer relay. A router represents a network layer relay, and a gateway is any relay at a layer that is higher than the network layer. Nonhomogenity and a broad variety of differences between interconnected networks needs to be resolved by numerous functions performed by those relays in order to achieve connectivity. Some of those tasks to be resolved are: a) routing techniques; b) error control; c) flow control; d) user access control; e) close procedures; f) communication monitoring and traffic handling; g) statistics; and h) network efficiency to name but a few. The present invention provides an intelligent communication and applications server to provide connectivity between wireless and/or land-line networks.

Referring now to Figure 1, a communications system is shown and is generally indicated to by reference numeral 20. As can be seen, the communications system 20 includes an intelligent communication and applications server 22 including a server 24 and a backup server 26 interconnected via an Ethernet backbone 28. Workstations 30 are also connected to the intelligent communication and application server 20 by way of an Ethernet based network 32.

The intelligent communication and applications server 22 is connected to a land-line based (wired) network 34 on one side and a wireless packet data network (WPDN) 36 on the other side. The land-line based network 34 includes a

plurality of host computers 40, 42, and 44 respectively, connected to the intelligent communication and applications server 22 by way of an X.25 land-line packet data network 46. The intelligent communication and applications server 22 is also connected to additional host computers 50 (only one of which is shown) by way of a TCP/IP protocol network 52.

The WPDN 36 includes a WPDN switch 60 interconnecting the intelligent communication and applications server 22 with one of a plurality of WPDN base stations 62. Wireless mobile terminals 64 communicate with the WPDN base stations 62 by way of wireless RF communications links 66. The wireless mobile terminals 64 are preferably of the point-of-sale (POS) / electronic fund transfer (EFT) type such as those described in PCT application No. PCT/CA96/00104 for an invention entitled "Free-Roaming Hand-Held Point-Of-Sale Terminal".

Referring now to Figures 2 and 8, the intelligent communication and applications server 22 is better illustrated. As can be seen, in Figure 2, the intelligent communication and applications server 22 includes a front end in the form of a virtual host 80 communicating with the WPDN 36 and a back end in the form of a virtual terminal 82 communicating with the land-line based network 34. A virtual gateway 84 connects and transfers data between the virtual terminal 82 and the virtual host 80 and accesses a database 86.

Turning now to Figure 8, the gateway 84 is better illustrated. Gateway 84 includes a message recognizer 100 receiving input data from the WPDN 36 and land-line network 34. The message recognizer 100 communicates with a knowledge base 102 (database 86 in Figure 2) and conveys logical messages to a message dispatcher 104. Message Dispatcher 104 conveys logical messages to an autorouter and autobridge 106 which in turn outputs data to the WPDN 36 and land-line network 34. The knowledge base 102 also communicates with an expert system tool kit 108 to allow the knowledge base to grow to meet the needs of the communications system 20.

Referring now to Figures 3 to 5, one of the mobile terminals 64 is schematically illustrated. As can be seen, the mobile terminal 64 includes a message

dispatcher 200 connected to a communication manager 202 and to the various components 204 of the mobile terminal 64 via a software bus 207. The message dispatcher 200 also communicates with a message office 206 including an internal registry 208 and mailboxes 210. The communication manager 202 communicates with a knowledge base 212 providing physical mapping. The communication manager includes an autobridge and autorouter 214, virtual ports 216, input/output device drivers 218 for outputs such as an RF modem, SPI bus, RS 232, etc. and a check registry 220.

Figure 7 illustrates the network infrastructure of the communications system 20 showing the mobile terminals 64 connected to the host computers 40 to 44 and 50 with the intelligent communication and applications server 22 acting between them.

The mobile terminals 64 communicate with the WPDN 36 which may be of the DataTAC, Mobitex, CDPD, GSM or PCS type. DataTAC wireless packet switching data networks require 56 to 64 kbps backbone communication pipe per physical connection with SCR (standard context routing) over X.25, through PVCs (permanent virtual circuits) or SVCs (switched virtual circuits). The intelligent communication and applications server 22 provides full connectivity to the host computers 40 to 44, 50 via this communication path, especially in the case where the mobile terminals transmit financial transaction data where ETE POS (end to end point of sale) protocol is a must. DataTAC engine servers (and others) do not provide ETE connectivity.

Mobitex wireless packet switching data networks are very similar to DataTAC WPDNs and therefore, the intelligent communication and applications server 22 must bridge the connectivity gap. CDPD wireless packet data networks are the most demanding in terms of throughput and speed. These networks are entirely based on the TCP/IP protocols. GSM/PCS networks are circuit switched wireless cellular networks where again end to end (ETE) connectivity is an issue.

The intelligent communication and applications server 22 is designed to provide full communication connectivity with the WPDN 36 regardless of its form

and any applications (i.e. mobile terminals 64) on the WPDN 36. The intelligent communication and applications server 22 is also designed to provide communication connectivity among the host computers 40 to 44 and 50 as well as communication connectivity among multiple host computers to multiple networks.

In the communications system 20, there are two basic groups of participants. One group of participants is constituted by the mobile terminals 64 which follow a logical model consistent with the intelligent communication and applications server 22. The second group of participants relates to the host computers 40 to 44 and 50. The host computers 40 to 44 and 50 do not follow the logical model followed by the mobile terminals 64 and intelligent communication and applications server 22. Thus, the intelligent communication and applications server is based on communications between a "known application" (i.e. a mobile terminal 64) and an "unknown host" (i.e. a host computer 40 to 44, 50). "Unknown" in the context of the present application refers to the fact that only the basic API level of the host computer is known to the intelligent communication and applications server 22. Since the host computers 40 to 44, 50 are typically managed by financial institutions, there is nothing that can be done to change the API level of the host computers in order to allow them to communicate directly with the mobile terminals 64. Therefore, this poses a potential communications problem especially when a message is transmitted from a host computer that is to be delivered to a mobile terminal 64.

According to Tanenbaum, every communication entity within a communications environment has to be identified in order to establish communications with other entities. Three terms that are applicable to this concept are names, addresses, and routes. The important step in this concept is the logical address which leads to the concept of an API logical message generated on an API level.

In the present communications system 20, the above-identified potential communications problem between host computers and mobile terminals is overcome by the intelligent communication and applications server 22 which builds API logical messages and wraps them with API data generated by the host computers.

In this manner, two way logical communications between the host computers 40 to 44, 50 and mobile terminals 64 is established. One special case that arises in communications between the host computers and the intelligent communication and applications server 22, is a result of protocols with "zero messages". Protocols of this nature do not comprise API data preventing the logical message from being built. This is solved by the intelligent communication and applications server 22 using the knowledge base 102.

To achieve communications in the communications system 20, participants (component) taking part in the communications traffic (including third party hosts etc.) have to be marked with a unique logical ID. The logical address format comprises three levels of identities, namely a system ID, an application ID and a component ID. The general format for the logical address is:

System ID. Application ID. Component ID.

The system ID is a unique ID of the system within the logical model. The length of the field (i.e. the number of bytes) depends on inputs and designates many of the systems. The application ID is the field of the logical address that represents every logical unit (application) within the communications system 20, such as the mobile terminals 64, the intelligent communication and applications server 22, etc. Basically any participant in the communications system 20 is assigned an application ID. This is an abstract definition which does not care about physical connections. The component ID field represents all possible components within all of these logical units (application) such as drivers, communication managers, bank credit components, bank debit components, etc.

The logical model enables logical connectivity among everything participating in the communication traffic. It is important to note that this connectivity is not physical connectivity. This is API logical connectivity established in order to enable application data to flow freely within the communications model. The intelligent communication and applications server 22 participates in this traffic bridging different applications of the system. Real physical mapping occurs within each of the applications (devices) as part of the knowledge base 212. Specifically, the

knowledge base 212 maps ports 216 and device drivers 218 on the mobile terminals 64 to ports on the intelligent communication and application server 22 that communicate with the X.25 packet data network 46, or TCP/IP protocol network 52 or Ethernet network 32. Figure 6 shows the connectivity of the logical model and Figure 3 shows component layouts for logical messages. The registration process is on-line for the mobile terminals 64 and off-line for the host computers 40 to 44 and 50.

In addition to logical addressing, data exchange between the two APIs is another important part of the logical message. Data can be exchanged in two manners, namely asynchronously or synchronously. The logical message indicates which communication manner is supported and embeds this information into its header as a Type and Tag.

Asynchronous data exchange is datagram oriented. During asynchronous data exchange, a request datagram is sent to the intelligent communication and applications server 22 by the application. The intelligent communication and applications server 22 responds to this request by returning a respond datagram. Recognition of the "Request-Respond" handshake in the logical message is done in the Tag field and is referred to as the handshake ID. The handshake ID is a unique ID and is returned by the intelligent communication and applications server 22. The handshake ID value is within the range of 1 to 65535.

Synchronous data exchange is session oriented. During synchronous data exchange, a request message (it can be as the first message of the conversation) to open a "conversation" session is sent to the intelligent communication and applications server 22 by the application. The intelligent communication and applications server 22 creates a session and its session ID is returned. During the conversation time via this session, both the application and the intelligent communication and applications server have to keep the same session ID until either the application decides to close the conversation or there is some communications problem from the server site. The session ID value is within the range of 1 to 32767

(2 bytes), where value 1 is a request to open a session and a negative value is a request to close a session. The value 0 is an invalid session ID.

The logical message format is as follows:

SID	Type	Tag	Src	Dst	API Message Body

SID represents a signature ID, that is calculated (CRC) from its header context. The field is used to recognize third party messages. The remaining fields support logical connectivity and the API data exchange mechanism.

Type is a message type indicator to recognize a data exchange mechanism. It reflects the manner between which two API components will cooperate.

Tag represents additional run-time information regarding to the type of the logical message. This tag ID is a unique conversation number between the two API components.

The datagram communication mechanism is as follows:

Tag ID = 0, means invalid session ID; and

Tag ID = 1 < Handshake ID < 65535.

The session (message) communication mechanism is as follows:

Tag ID = 0, means invalid session ID;

Tag ID = 1, means request to open session, or the first message;

Tag ID = 1, < Session ID < 32767 means a valid session; and

Tag ID = - 32767 < Session ID < -1, request to close a session, or the last message.

Src. is a logical address of the source and Dst. is a logical address of the destination.

The API message body has the following format:

EXT	**D/M**	**Duplicate**	**Reserve**	**Reserve**	**b2**	**b1**	**b0**

Ext is an extension bit to allow insertion of another header;

D/M is the datagram or message;

Duplicate is a logical message duplicate; and

b2, b1 and b0 are bits describing data exchange.

For this logical connectivity concept there is a one special service within the communication devices in the communications system 20, namely, the registry databases. The logical connectivity concept between the components in the client/component configuration is based on "unknown information" of the component location. The application's component can be moved to any place within the logical model without requiring changes to API level to be made.

Each registry database is a knowledge base of the connectivity of the internal and external components of the application. Every internal component of the application has a duty to perform a registration and unregistration process service. In the case of the mobile terminals 64, the message dispatcher 200 manages this process service. Based on component registrations, the message dispatcher 200 can simply decide if an API message is to be performed internally or whether it is necessary to pass the API message to the communication manager 202. The communication manager 202 handles the incoming and outgoing logical messages (part of the API message).

There is a second part of the registry database dealing with the information necessary to route the message. This database is the knowledge base 212 dealing with the physical mapping between the physical device and the logical destination address. There are two types of registry databases. The first type is controlled by a component on the "fly" (run-time registration) and it does not need to

be administrated. The other type is for all external connectivity and it needs to be administrated.

The registry database stores the logical connections to or from the internal components with the mapping to the physical communication ports. Each mobile terminal 64 therefore has to have only one database with a minimum of one logical connection to the administrator of the system to achieve connectivity. The image of the external database can be constructed or updated in different ways and by different sources. For example, the external database can be constructed or updated (i) locally from the application manager; (ii) remotely from the administrator; or (iii) on the "fly" (run-time) as a learning process.

The registry database is a knowledge base component - communication abstract. The message dispatcher 200 and communication manager 202 manage message flow. They also have private properties to deal with how to handle message traffic (all kinds of information example - Host protocol description which is important with TTY host protocols which have "zero messages"). This can be embedded in the registry database.

Figures 4 and 5 show the internal outgoing and incoming message control flow within the mobile terminals 64. For an outgoing message, a component 204 sends a message that goes to the message dispatcher 200. Within the message dispatcher, a check is performed within the check registry that communicates to the message office 208. All internal messages stay within the mobile terminal 64. External messages go to the communication manager 202 which contains the external registry 212 and autorouter 214. Autorouter is responsible for relaying the message to the appropriate device driver port 218 (RF modem, RS232, etc.).

For incoming messages, the message arrives via the RF modem or RS 232 port and passes the check registry 220 in the communication manager 202. After confirmation, the incoming message goes to the message dispatcher 200 for the next level of check and through the store manager mechanism before the message is stored in the mail box 210. Any component 204 can get a message from the mail box.

The logical connectivity concept embodied in the communications system 20 including the logical address and the API logical message is the abstract core that shapes the internal architecture of the intelligent communication and applications server 22.

In the case of the intelligent communication and applications server 22, logical messages (as a part of the API header) that come from all communications components, devices and all traffic participants proceed to the message recognizer 100. Messages that originate from the mobile terminals 64 can bypass the message recognizer 100 and be routed to the proper destination. Messages that come from other sources pass through the message recognizer 100. In the case of a message from an "unknown" host computer, the logical message does not exist within the message header, so it has to be built by the message recognizer 100. The message recognizer 100 uses fuzzy logic and artificial intelligence to do this. In order to build the logic message header, knowledge from the knowledge base 102 is used. The knowledge base is initially built up and updated via the tools. The tools 108 serve to enter basic API requirements of the "unknown" host computers. The knowledge base 102 builds itself (using complex algorithms) as the traffic between mobile terminals 64 and host computers 40 to 44 and 50 progresses.

Figures 9 and 10 show mobile terminal 64 to host computer 40 to 44, 50 traffic and host computer to mobile terminal traffic while Figure 11 shows one example of the OSI model protocol stack and its conversions in the communication chain, based on a DataTAC 5000 (Motorola) wireless packet data network. Application is POS/EFT Datapack 3201 standard.

The intelligent communication and applications server 22 includes virtual host wireless communication defaults. One of the few defaults is a DataTAC core. The other defaults are CDPD, GSM/PCS cores. This is the front end of the server. All of these cores communicate with one common core.

In addition, many wireless networks provide for "fleet connectivity" (one to many). Fleet connectivity cares only about the destination. This is a synchronous protocol. Between the intelligent communication and applications server

22 and the WPDN switch, messages are routed via SVC's (switchable virtual circuits) located in the so called SVC SINK. SVC concept comes from the packet data networks (like X.25). The intelligent communication and applications server 22 and the host computers communicate synchronously via dedicated SVCs. That means that a limited number of SVCs have to handle many mobile terminals. This traffic is handled by the intelligent communication and applications server 22.

Server 22 enables three types of connectivity through the SVCs and enhances communication throughput, namely device connections, sessions connections and asynchronous datagram messaging. Asynchronous datagram messaging follows the dynamics of the traffic and accommodates any increase in the traffic. A limited number of SVCs provide throughput for a much larger number of mobile terminals. Device connection is permanent-static (terminal corresponds to the specific SVC). Session locks communication SINK only temporarily.

Although global logical connectivity is important for the message delivery aspect of this invention, the intelligent communication and applications server 22 functions not only as a communication bridge, but also serves to execute and generate other API message bodies or to convert applications into different forms.

Although a preferred embodiment of the present invention has been described, those of skill in the art will appreciate that variations and modifications may be made without departing from the spirit and scope thereof.

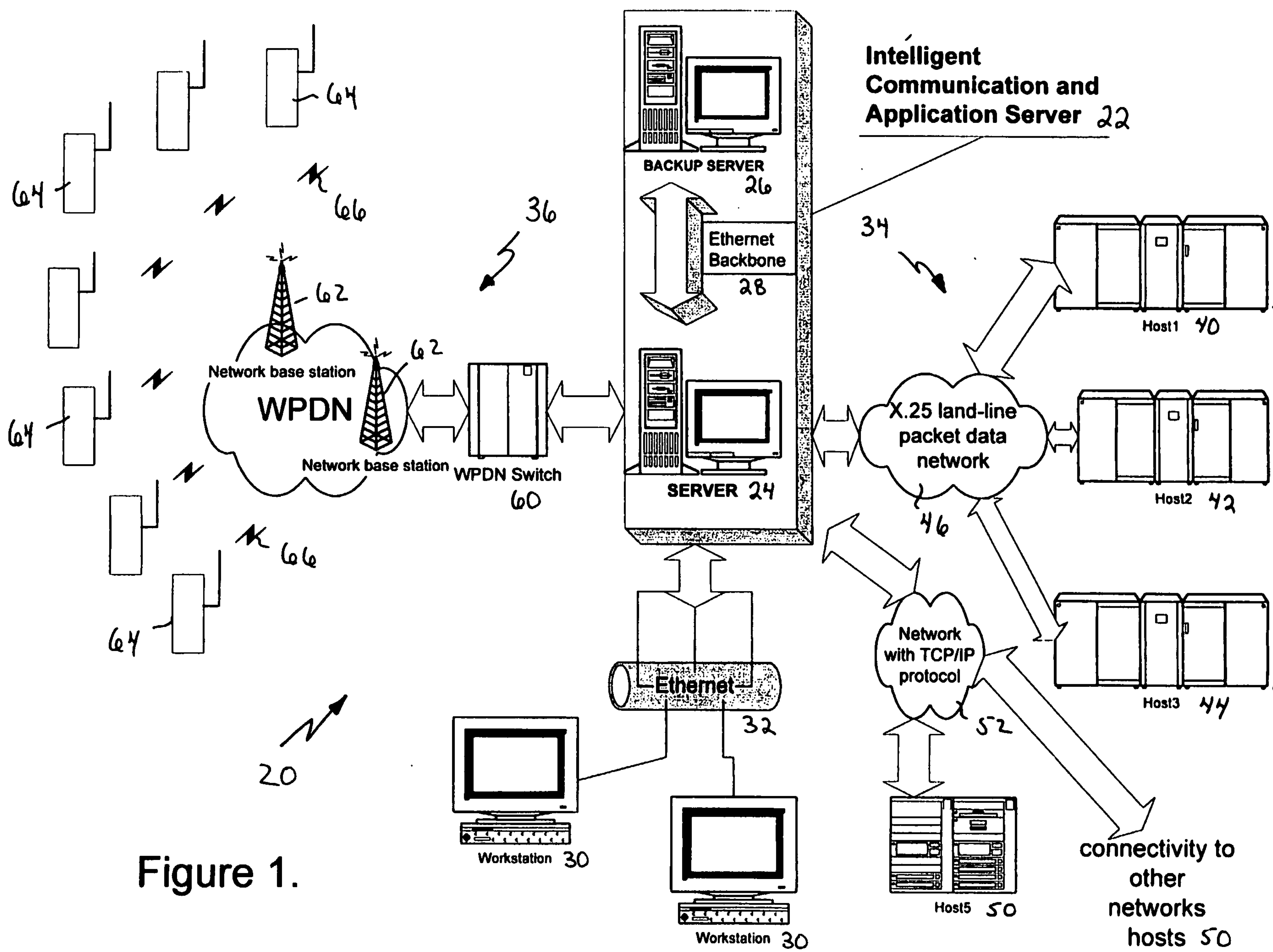
Intelligent Communication and Application Server 22
BACKUP SERVER 26
Ethernet Backbone 28
SERVER 24
36
34
Network base station
WPDN
Network base station
62
62
64
64
64
64
66
66
WPDN Switch 60
X.25 land-line packet data network
46
Host1 40
Host2 42
Host3 44
Network with TCP/IP protocol
52
Host5 50
connectivity to other networks hosts 50
Ethernet
32
Workstation 30
Workstation 30
20

Figure 1.

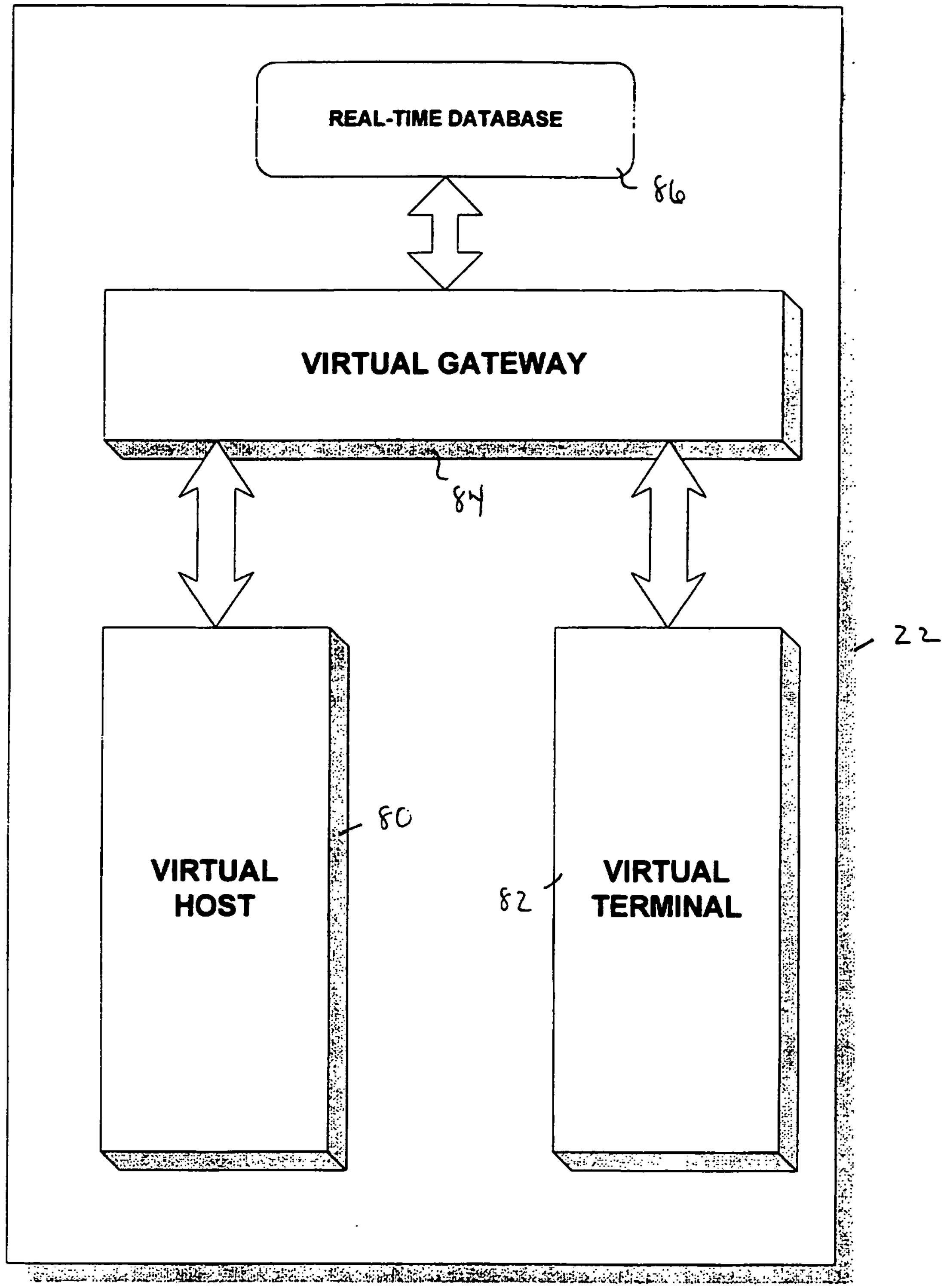
REAL-TIME DATABASE
86
VIRTUAL GATEWAY
84
22
VIRTUAL HOST
80
82
VIRTUAL TERMINAL

Figure 2.

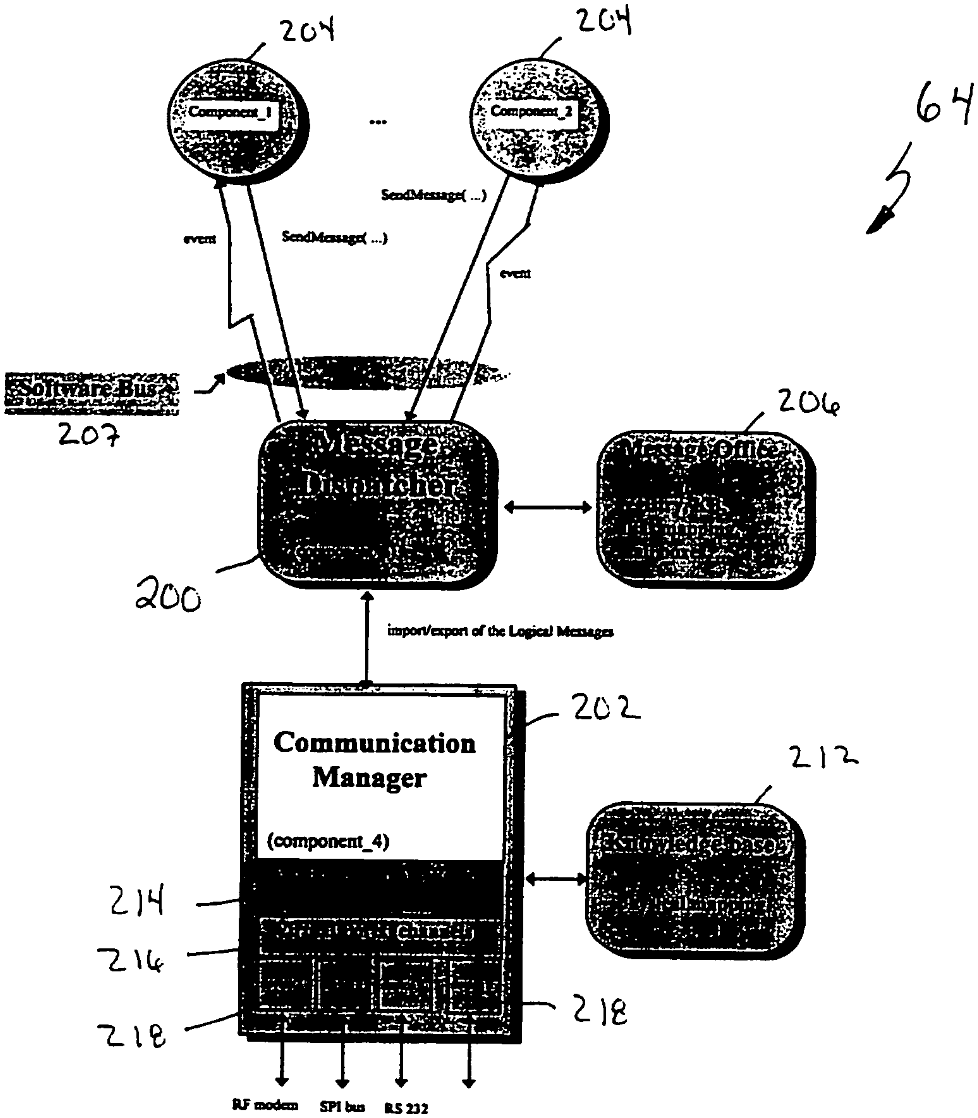

Figure 3. The communication componnents within the physical device.

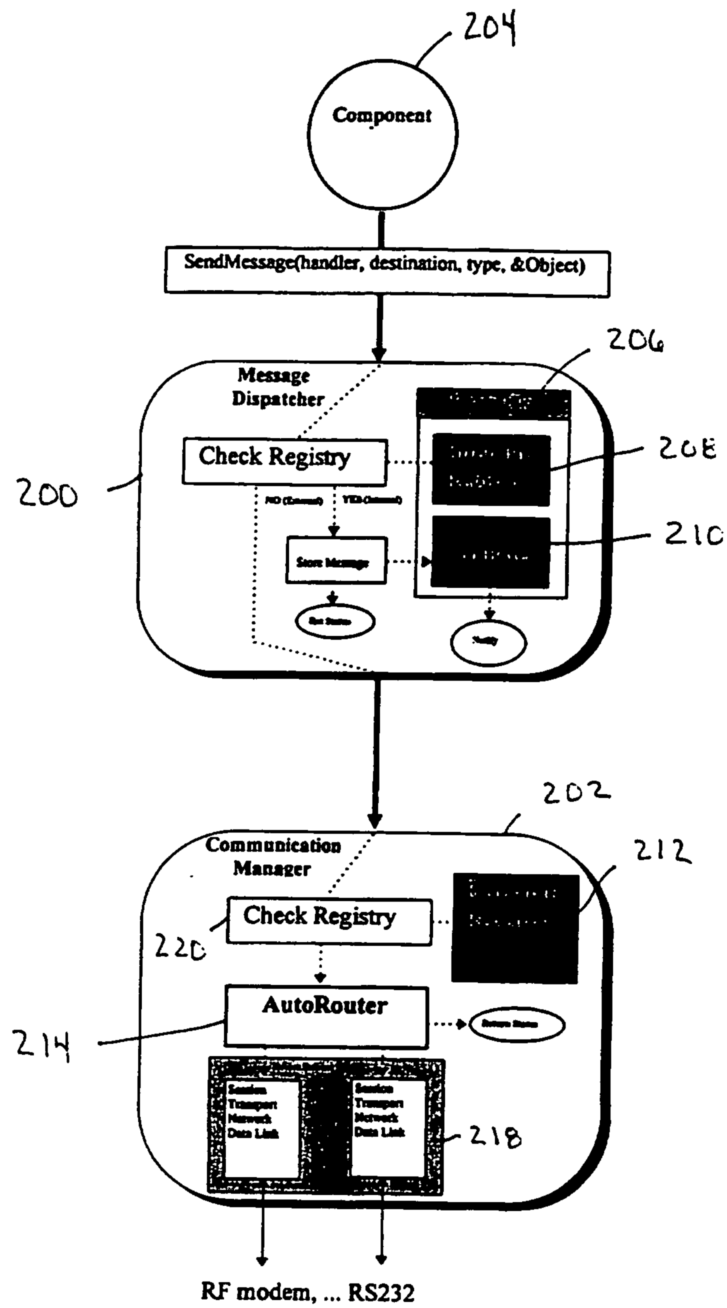
204
Component
SendMessage(handler, destination, type, &Object)
206
Message
Dispatcher
Check Registry
208
200
210
Store Message
202
Communication
Manager
212
Check Registry
220
AutoRouter
214
Session
Transport
Network
Data Link
Session
Transport
Network
Data Link
218
RF modem, ... RS232

Fig. 4.

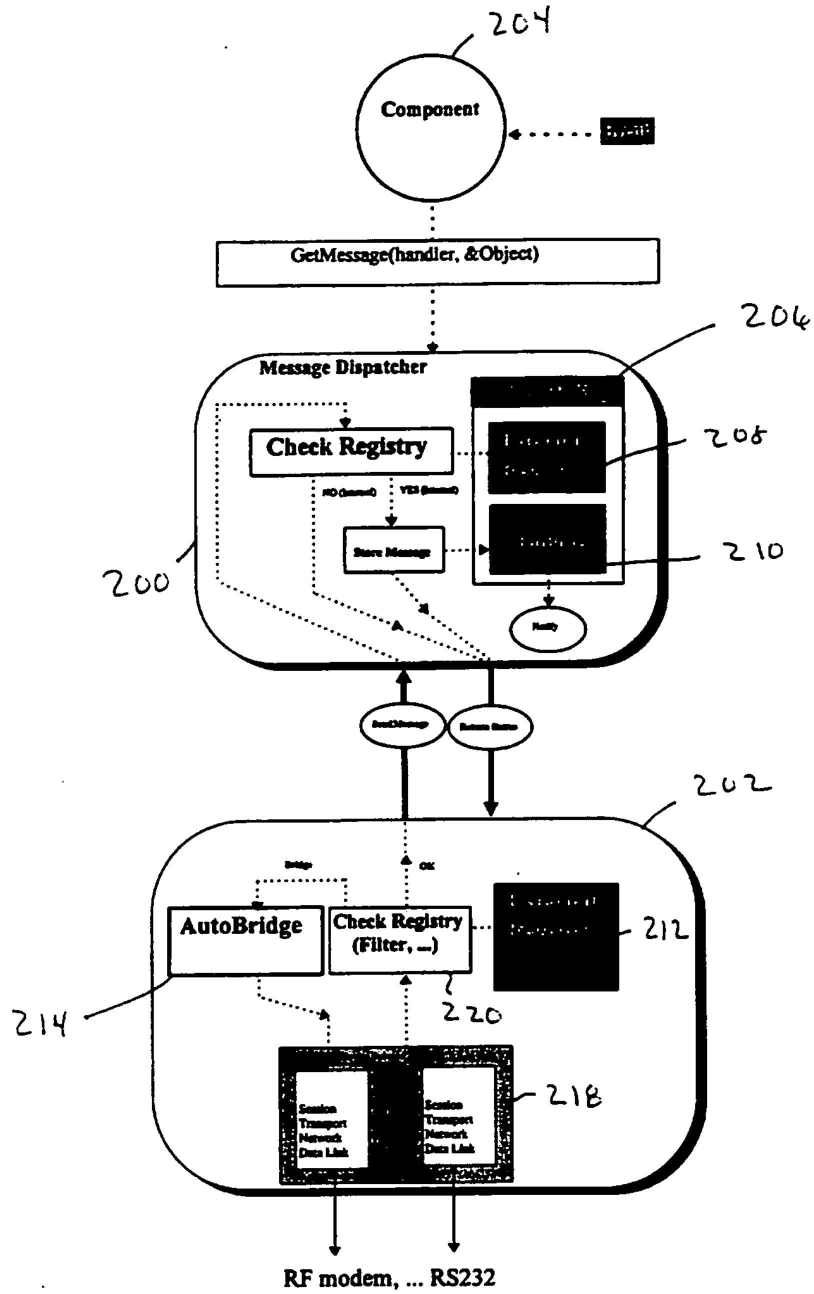
204
Component
GetMessage(handler, &Object)
206
Message Dispatcher
Check Registry
208
Store Message
210
200
202
OK
AutoBridge
Check Registry
(Filter, ...)
212
214
220
Session
Transport
Network
Data Link
Session
Transport
Network
Data Link
218
RF modem, ... RS232

Fig. 5.

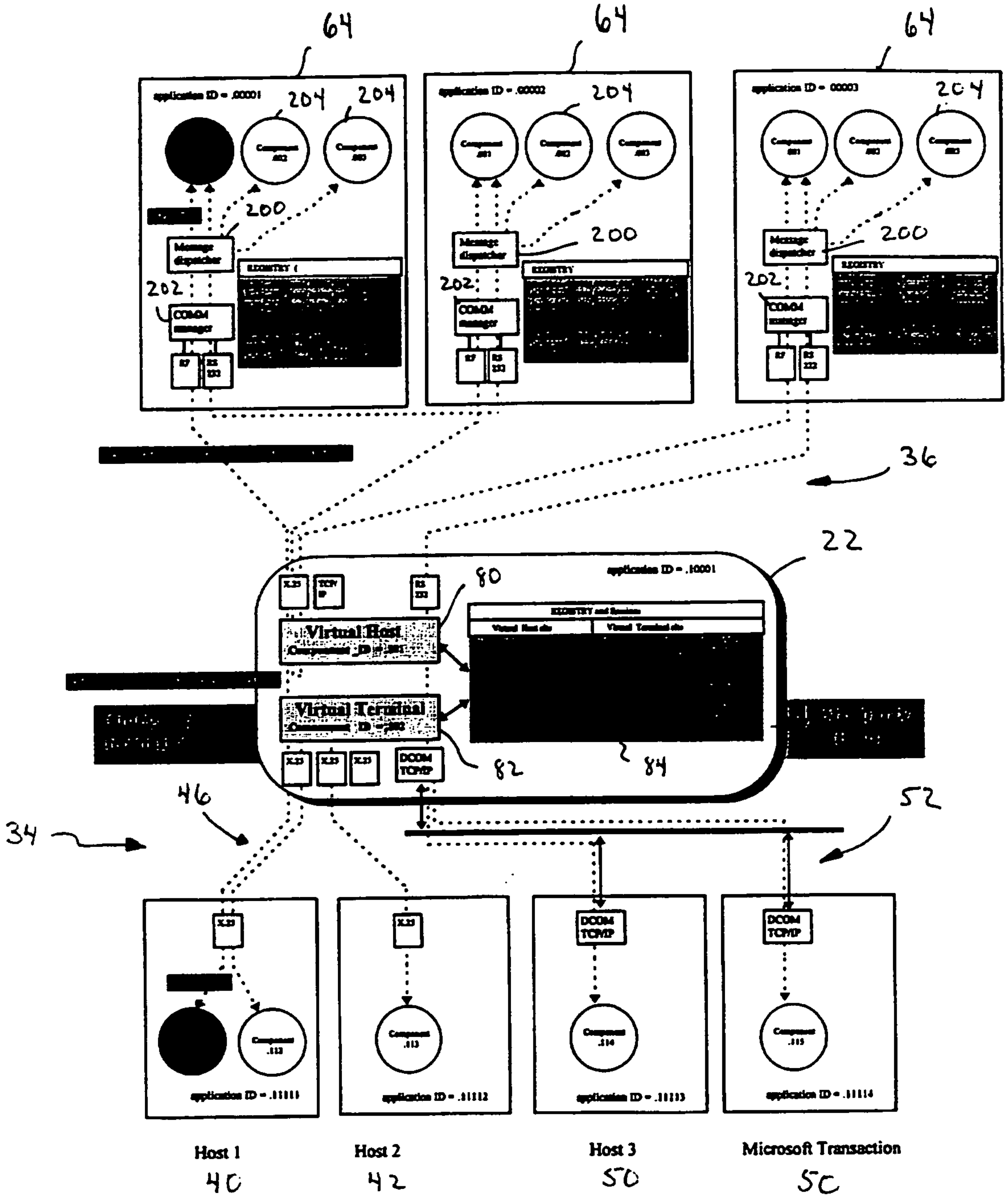

Figure 6. The Connectvities of the Logical Model - systemID = 0001

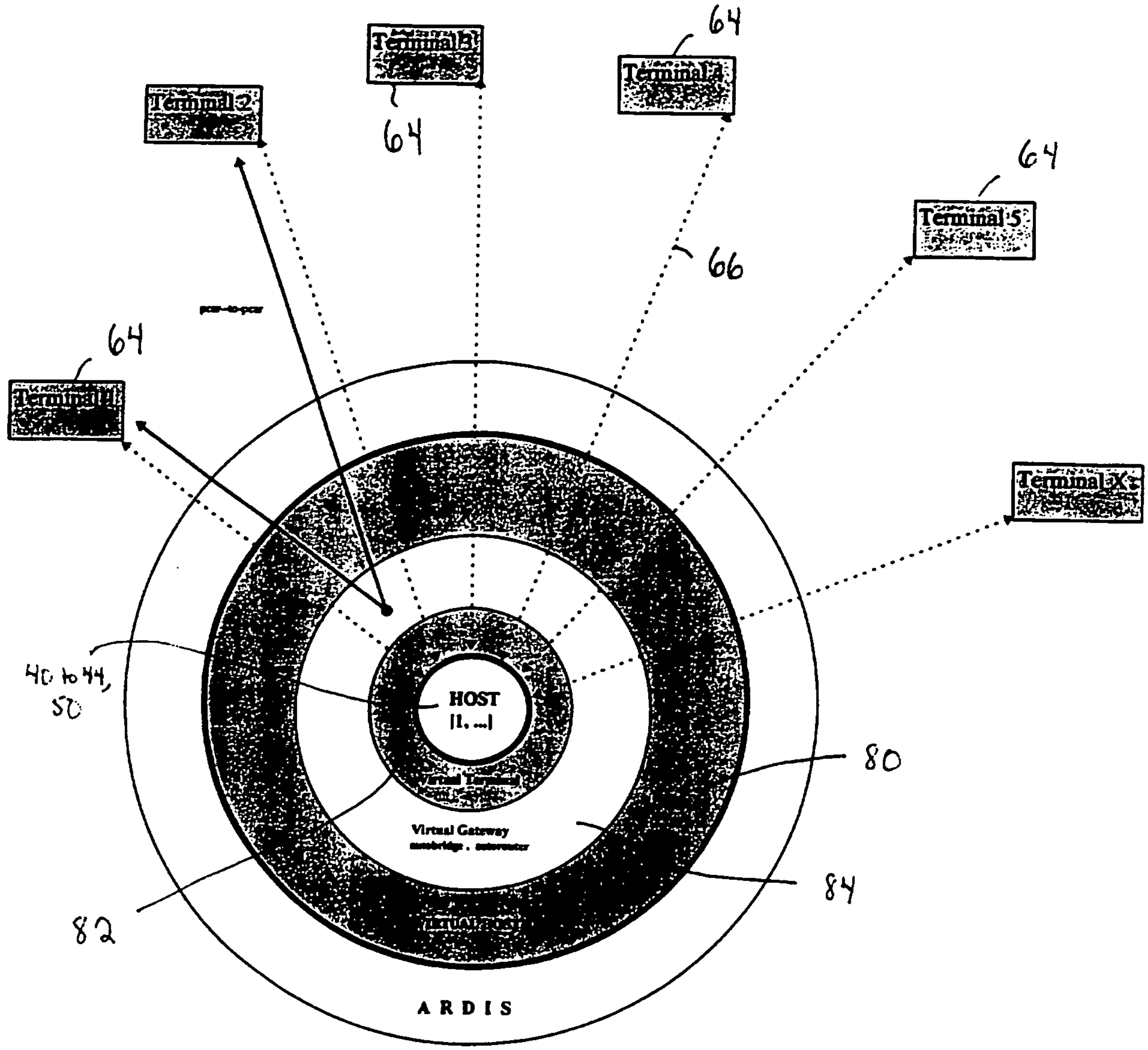

Figure 7. Terminal to Host Network Infrastructure

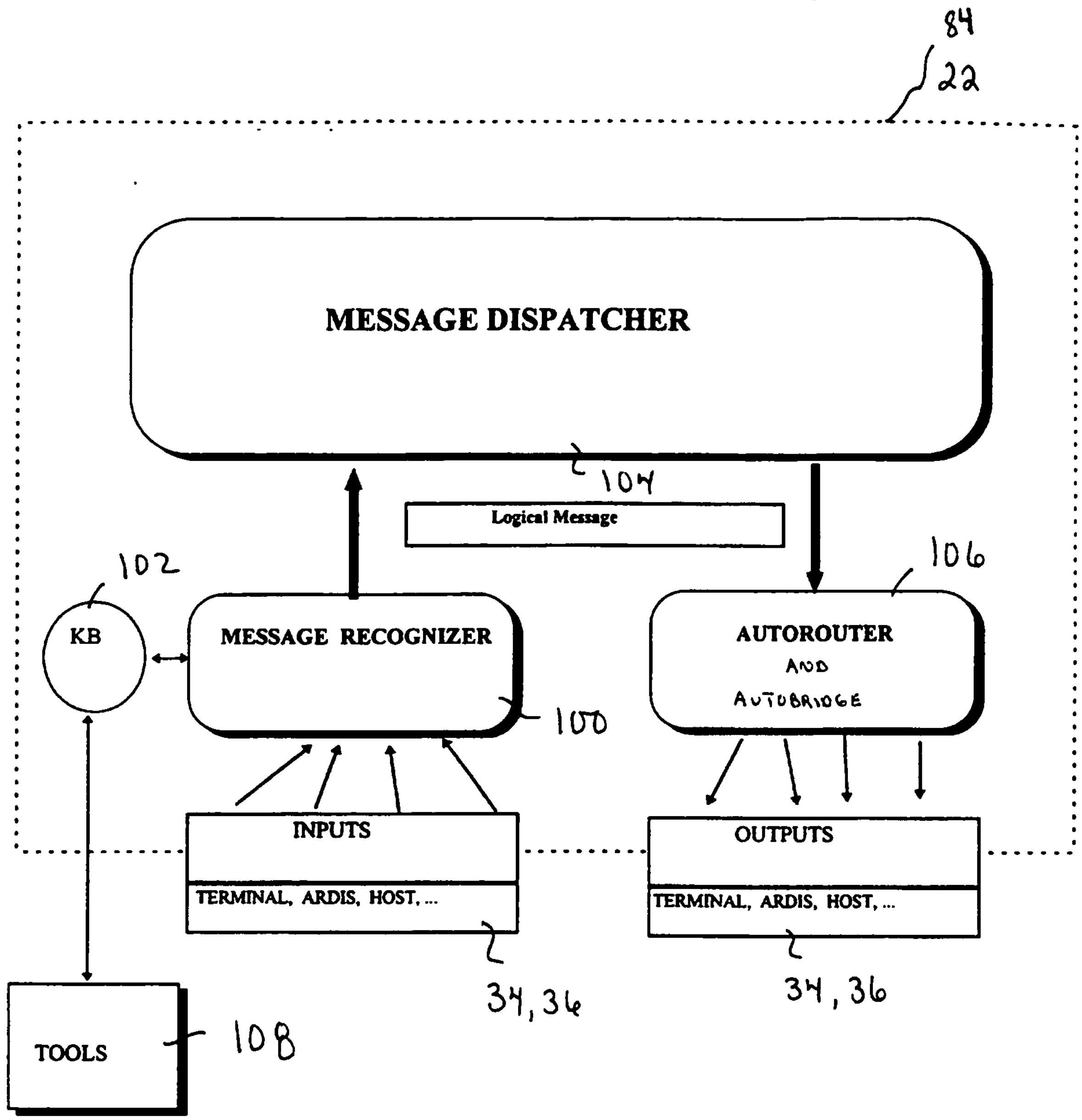
84
22
MESSAGE DISPATCHER
104
Logical Message
102
KB
MESSAGE RECOGNIZER
100
AUTOROUTER
AND
AUTOBRIDGE
106
INPUTS
TERMINAL, ARDIS, HOST, ...
34,36
OUTPUTS
TERMINAL, ARDIS, HOST, ...
34,36
TOOLS
108

Fig. 8

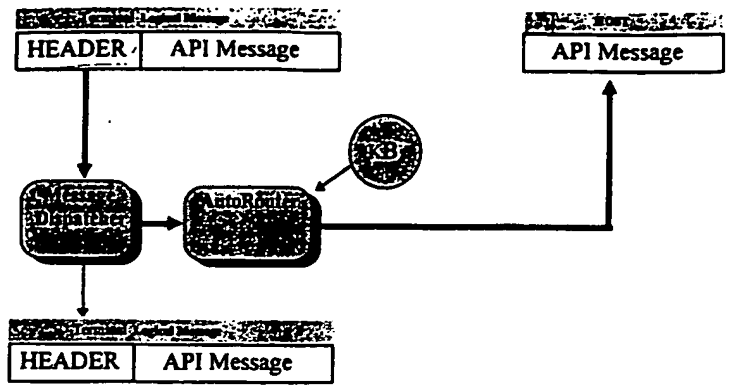

Figure 9. Terminal to Host Traffic

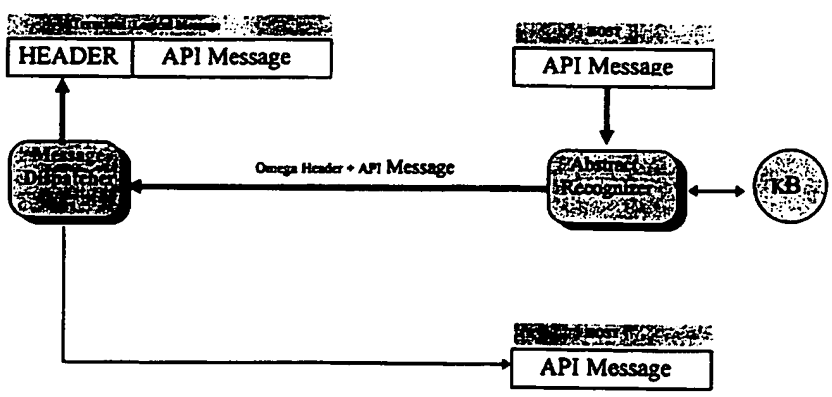

Figure 10. Host to Terminal Traffic

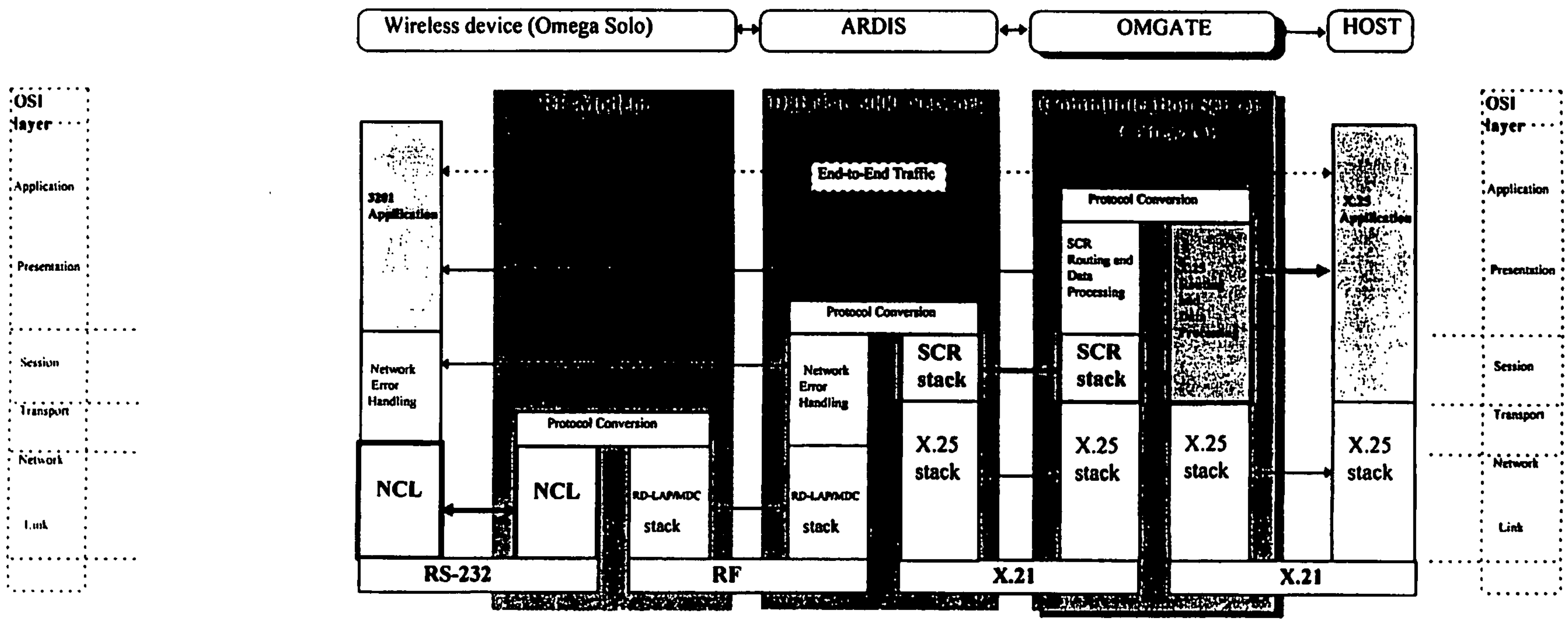
Wireless device (Omega Solo)
ARDIS
OMGATE
HOST
OSI layer
Application
Presentation
Session
Transport
Network
Link
3201 Application
Network Error Handling
NCL
Protocol Conversion
NCL
RD-LAP/MDC stack
End-to-End Traffic
Protocol Conversion
Network Error Handling
RD-LAP/MDC stack
SCR stack
X.25 stack
Protocol Conversion
SCR Routing and Data Processing
SCR stack
X.25 stack
X.25 stack
X.25 Application
X.25 stack
RS-232
RF
X.21
X.21

Fig. 11.